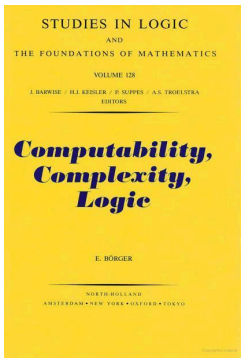
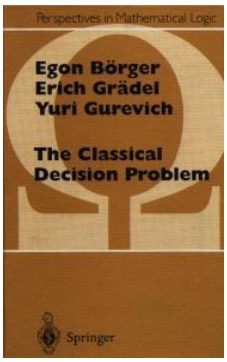
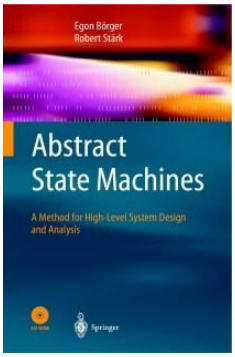
Egon Börger (Pisa): **Does the Science of Computing have anything to do with Logic?**

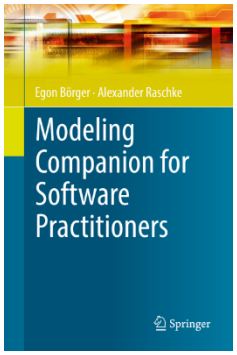
(Retrospective view of 50 years under the spell of computer science)

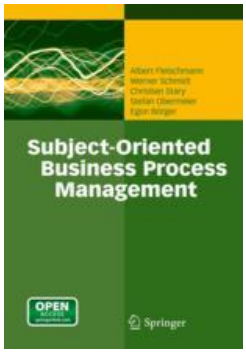
Three stages characterize my life in science which started in philosopy and through mathematical logic (the first two decades) brought me to the science of computing (the last three decades):











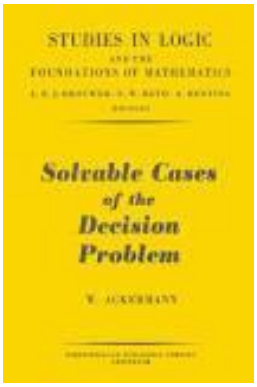
Paraphrasing Bertrand Russell one could say that I dedicated myself as a young man to mathematics and logic, as an adult to software engineering and in my later years to modeling business processes. Honit soit qui mal y pense.

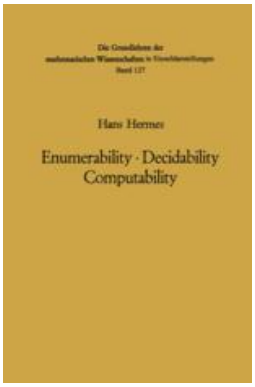
I actually wanted to become an orchestra conductor but my music teachers warned me that conductors were a dime a dozen. Instead, they suggested I should become a pianist or an organist, none of which appealed to me. Therefore, I decided to study philosophy. Owing to my school background in humanities, I wanted to understand the fundamental principles that govern the world.

The lectures at the **Sorbonne** in Paris and at the Institut Superieur de Philosophie in **Louvain**, and in particular those held by professors Joseph Dopp und Jean Ladriere https://hiw.kuleuven.be/claw/about who followed the tradition set by Robert Feys, piqued my curiosity in mathematical logic. This brought me in 1966 to the Institut für Mathematische Logik und Grundlagenforschung in Münster, Germany, and hence back to my hometown – the one place where I hadn't wanted to pursue my studies and certainly not reading mathematics!

Once there, I was taken in by the spirit of the institute of that time: its founder Heinrich Scholz https://de.wikipedia.org/wiki/Heinrich_Scholz_(Logiker) as well as his student and successor Hans Hermes https://de.wikipedia.org/wiki/Hans_Hermes belonged to the few who had promptly recognised the epoch-making significance of Turing's solution to Hilbert's fundamental Decision Problem, were aware of the intimate connection between abstract machines and logic languages to describe them, and were interested in the further development of such (as we say nowadays `formal') languages (see Rainer Glaschick: Alan Turings Wirkung in Münster, Mitt. Dtsch. Math.-Ver. 2012, 20 (1), 42-48).

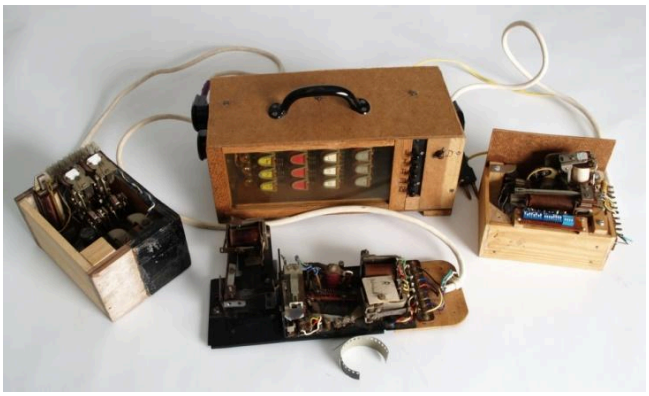
The resulting **Münster Tradition** of a **computational focus on logic** was developed further by Gisbert Hasenjaeger https://de.wikipedia.org/wiki/Gisbert_Hasenjaeger, Hans Hermes (see his influential textbook on "Aufzaehlbarkeit, Entscheidbarkeit und Berechenbarkeit"), and Dieter Rödding https://de.wikipedia.org/wiki/Dieter_R%C3%B6dding. Also Wilhelm Ackermann's lectures https://de.wikipedia.org/wiki/Wilhelm_Ackermann_(Mathematiker) in Münster belong in this list, and especially his influential book about "Solvable Cases of the Decision Problem" (1954), written in Münster.

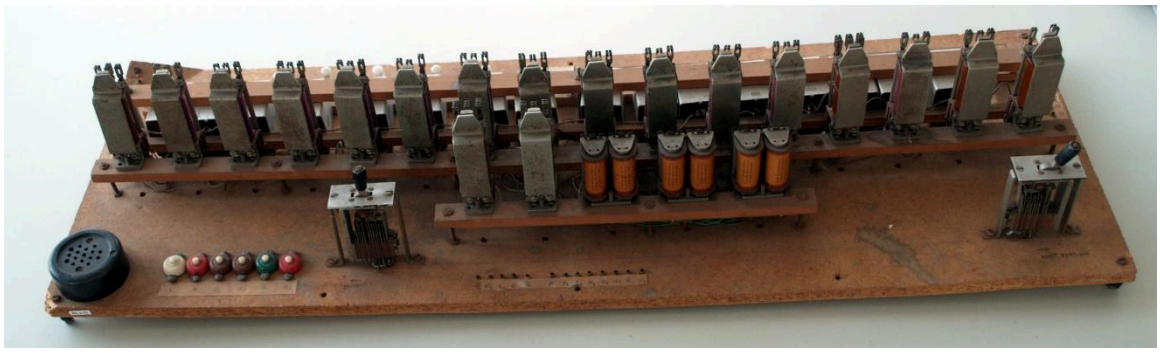




Since 1966, in Rödding's lectures and seminars you could not just learn the classic themes of mathematic logic – logic languages and calculi for predicate logic, type theory, modal logic, set theory, model theory, proof theory, etc. – but you would equally receive also an intensive exposure to many concepts of "computing machines", like networks of finite automata, cellular automata, Petri nets, Chomsky hierarchy automata, Turing machines, register machines, machines that work on complex data structures (Sic), Lambda and Fitch calculations, combinatory logic, Markov algorithms etc. Paramount, reflecting the spirit of Turing, was the study of algorithmic complexity of decision problems and of hierarchies of machine-computable functions.

In the Turing room, set up by Hasenjaeger and further developed by his student Rödding (but disbanded shortly after Rödding's death), the automata explained during the lectures, even small universal Turing and register machines, were built with the simplest of means and presented as part of exercises. See here a picture of Hasenjaeger's MiniWang, a small universal Turing machine, next to what probably is a counter, that together with other artefacts were saved by Walburga Rödding from the dismantling of the Turing room. These items are currently displayed together with Hasenjaeger's estate in the Heinz Nixdorf MuseumsForum in Paderborn http://www.hnf.de/start.html
where they expect further investigation.





That this work by no means lacked its uses is shown by the article ‘‘Wang's B machines are efficiently universal, as is Hasenjaeger's small universal electromechanical toy “ by T. Neary, D. Woods, N. Murphy und R. Glaschick (J.of Complexity 30 (2014) 634-646 and https://arxiv.org/abs/1304.0053)

as well as by the concept of register machines, developed independently and at the same time by Minsky, Sheperdson and Sturgis, and Rödding (see Hasenjaeger: "On the early history of register machines", LNCS 270, 181-188).

How I managed as student and assistant not to have to help with soldering in the Turing room remains a mystery to me. But I have to thank this machine-focussed mathematical training at the logic institute in Münster for my interest in the science of computing that was just about to emerge as a scientific discipline at the time.

Thus, one year after my PhD with a dissertation about the complexity of logical decision problems, I accepted the invitation of the physicist and cyberneticist Edoardo Caianello https://it.wikipedia.org/wiki/Eduardo_Caianiello to help set up an institute for computer science as lecturer at the University of **Salerno**, Italy (1972-1976). This is where, by giving lectures (!) about the various sections of information technologies, I acquired my technical knowledge of the discipline. After having completed my Habilitation in Münster in 1976 with a dissertation on algorithmic complexity measures, followed there by two years of lecturing in logic (1976-1978), I systematically broadened my knowledge in computing lecturing at the computer science institutes of the Universities of **Dortmund** (1978-1985) and **Udine** (Italy, 1983/4). The insights that I gained strengthened my conviction that the essential new challenges for mathematical logic, and not just those deriving from the theory of algorithmic complexity, come from the foundations of the science of computing.

The relationships between logic languages and classes of machines determine the above-mentioned textbook about "Computability Complexity Logic", first published in German, encouraged by Rödding and written during my time in Dortmund, as well as the later monography "The Classical Decision Problem". In both books, like in my dissertation, register machines (!) play an important role.

With the aim to promote the cross-pollination between **Logic and Computer Science** also on an organizational level, I started to systematically organize summer courses and seminars (in total about three dozens) about the interdisciplinary theme of 'logic and computer science', **back then a strongly controversial subject in the German academic logic circles**. I also published over two dozens of anthologies, starting with the "Course on Computation Theory" that took place in 1984 at the CISM in Udine, Italy (with the revised lecture material published in 1988 by Computer Science Press as "Trends in Theoretical Computer Science"), the 1984 symposium on „Recursive Combinatorics" (Springer LNCS 171) organised together with Hasenjaeger and Rödding in Münster (internally known as the 3-generations conference), and the book "Computation Theory and Logic" (1987, Springer LNCS 270), edited in Pisa (Italy) in memory of Dieter Rödding.

To provide such meetings with an institutional frame, with Hans Kleine Büning (from the Münster logic school) and Michael Richter we realized the idea of an annual "Computer Science Logic (**CSL**)" conference, launched following the sudden death of Dieter Rödding. This was in 1985, one year before news of the newly established "Logic in Computer Science" (**LICS**) conference crossed the Atlantic. The fact that the CSL-Series did not already start in 1986 and not at the logic institute in Münster, but only in 1987 and at the Institut fuer angewandte Informatik und Formale Beschreibungsverfahren in Karlsruhe was due to some adverse local interests in Münster and my move to the University of Pisa that these caused. In 1992, during a Dagstuhl seminar attended by logicians and computer scientists from 14 countries I proposed to transform the CSL conference series into the annual gathering of a European Association for Computer Science Logic (**EACSL**). The first of these meetings took place in San Miniato (Pisa) that year (s. https://eacsl.kahle.ch/goals.html). Together with LICS, the CSL conference remains to this day a flagship in theoretical computer science, see the report

"Ten Years of CSL Conferences (1987-1997)"

http://www.di.unipi.it/~boerger/Curriculum/HistoryCslEacsl87-97.pdf in the EATCS Bulletin 63 (October 1997) about my activity as first EACSL chairman, as well as the current conference list https://eacsl.kahle.ch/conferences.html

In order to get to know first-hand the challenges of creating correct industrial software systems, during the five sabbatical years I have been granted by the University of Pisa since 1985 I have worked at relevant companies (IBM, Siemens, Microsoft, SAP) as well as at the ETH Zurich. This experience has allowed me to see **practical applications of logical methods in software development** that I wouldn't have thought of; in particular

a) for the compilation of precise requirements documents, meant as technically and legally binding blueprints of the software systems about to be built, and
b) for their reliable, i.e. precisely documented and therefore objectively verifiable implementation by programs that can be executed by machines.

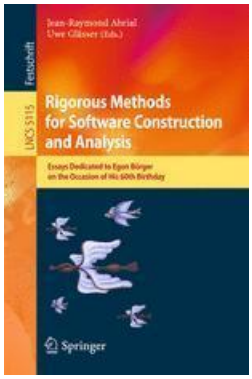
Point a) is about an epistemological problem that is composed of two parts:

- thorough *understanding* of desired system behavior (not only of equationally defined mathematical functions like those we as students of numerics had to program 50 years ago with the help of entire cases of punch cards),
- an unequivocal, objectively correct and watertight *formulation* of this understanding in formal (ultimately programming) languages.

This closes the circle to the precise analysis of natural language texts which we were intensively trained to produce during Dopp's exercise groups in logic in Louvain!

Point b) touches on a problem of applied model theory, namely the definition of model transformations that can be proven to be correct, especially via controlled (read: verifiably correct) gradual inclusion of design and programming decisions by so-called model refinements. The modern, abundant set of concepts and tools of *Computational Logic*, developed in reaction to the needs of computing, offers many practical, machine-supported methods for a mathematical *analysis* of software system models that complements their empirical test-based checks pragmatically and cost-effectively. Illustrative examples can be found in the books on Java/JVM, ASMs and the Subject-Oriented Business Process Modeling approach SBPM mentioned earlier on.

The adoption of methods borrowed from logic for the construction of verifiably correct software systems is also the main subject of a series of international conferences,
founded once again Dagstuhl in 1996



together with Jean-Raymond Abrial, Uwe Glaesser and other colleagues, held every other year, with the three-letter acronym **ABZ** https://www.southampton.ac.uk/abz2018/index.page .
Nomen est omen, as the ancient Latin saying goes. So here the initials of the methods involved: *Abstract State Machines, B, Z,* all of which rest on the foundation of mathematical logic and axiomatic set theory, albeit in different ways.

By now, over the past 50 years an unfathomable multitude of people and institutions the world over have developed a proper understanding of the manifold cross-dependencies of logic and theoretical as well as practical computer science which today are no longer disputed. *Pars pro toto*, from a European perspective I would like to mention two persons whose contribution is evident only to those in the know: Alfred Hofmann of the publishing house **Springer** who accepted to publish many early, and at the time controversial, LNCS volumes about interdisciplinary work in the overlapping area between logic and computing, as well as Reinhard Wilhelm for his foresight in supporting many influential **Dagstuhl seminars** about these multi-faceted themes.



The fertile development of *Computational Logic* and of the wide ramifications of practical applications of mathematical logic in the science of computing as well as, for me, the recognition of my contribution by the Alexander von Humboldt Research Award received from my home country in 2007, if looked at in a global perspective, have compensated for the damage inflicted upon mathematical logic in Germany by the short-sighted and in some quarters destructive attitude of a small group of its professors in the last quarter of the last century, a sort of scientific obstruction that also affected me personally. They had no care for the potential that computing held in store for their own discipline. I distanced myself from this circle by accepting in 1985 the invitation, ad personam, to join the computer science department of the University of Pisa.

NB. For the german original of this retrospective see Hat Informatik mit Logik zu tun?